ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing

HS 2015 Prof. R. Wattenhofer

Distributed Systems Part II

Exercise Sheet 4

Quiz

1 Zyzzyva

a) In Zyzzyva, the primary of a view which is ending continues taking part in the protocol in the next round as a replica. In reality we would like to get rid of byzantine nodes to improve performance. Imagine you are running Zyzzyva and notice that the system is slowing down. Can you unplug the primary and replace it with a new machine without making the system unsafe, i.e. without losing any complete commands?

b) Imagine that a byzantine client u cooperates with a byzantine primary p. The primary p orders commands requested by client u inconsistently across replicas. For example u requests two commands, $c_1 = v+1$ and $c_2 = v \cdot 2$. The order in which c_1 and c_2 are executed influences the final value of v. How does Zyzzyva detect and resolve this situation?

c) In the absence of failures, three rounds of communication are necessary to complete a command. Imagine the primary is correct but there are f byzantine replicas that slow down Zyzzyva as much as possible. How many rounds of communication are required to complete a command in this situation?

Basic

2 Zyzzyva . . . again

a) In Zyzzyva, replicas that initiate a view change by sending $\texttt{IHatePrimary}_r$ to all the other replicas do not stop participating in the current view until they collected $f + 1$ $\texttt{IHatePrimary}_r$ messages. Imagine replicas would immediately stop participating in the current view after sending $\texttt{IHatePrimary}_r$. Do you see how f byzantine nodes could sabotage the whole system such that no command can complete?

b) We have seen how during a view change, complete commands are recovered to construct a new history for the new view. Can a command that did not complete in the old view do so in the new view without the client acting at all after it requested the command from the primary?

Advanced

3 Authenticated Agreement

Algorithm 4.2 in the lecture uses authentication to reach agreement in an environment with byzantine processes.

a) Modify this algorithm in such a way that in handles arbitrary input. Write your algorithm as pseudo-code. The processes may also agree on a special "sender faulty"-value.

Hint: implement `value` as a set, work with the size of the set.

b) Proof the correctness of your algorithm.

Mastery

4 Even Faster Zyzzyva

We have seen how Zyzzyva relies on $3f + 1$ replicas to work fast in the absence of failures. As soon as one non primary replica is silent, every client must create a commit certificate for every command.

If the primary is correct, we would like to avoid assembling and broadcasting commit certificates for every command.

Assume that we now have $5f + 1$ replicas, i.e. the number of correct replicas is increased or the number of byzantine replicas is reduced.

a) Adapt Zyzzyva such that a command always completes in three communication rounds, if there are up to f byzantine replicas (assuming that the primary is correct).

b) Sketch a proof that the changes you made still lead to a safe system.